Course Project CCAI 413
Natural Language Processing
Third Trimester 2022/2023

Abusive Language Detection in Social Media

Authors: Abdulrahman khaled, Abdulaziz alamri, Ahmed Munir, Hazem Baik

Abstract

The report presents a study on the task of detecting abusive language in social media, the goal is to develop a machine learning model that can accurately classify tweets as abusive or normal. The study employs natural language processing (NLP) techniques and asses the baseline model performance. The results highlight effectiveness of the selected metrics and provide insight into the challenges in addressing abusive language detection.

Introduction

Our research focuses on the identification and management of offensive and hate speech exchanges on social media platforms, which has become a growing concern in recent years. Various studies have addressed this issue and proposed effective approaches. For instance, Smith et al. (2018) [1] conducted a comprehensive analysis of offensive language detection techniques, highlighting the importance of machine learning algorithms in identifying such content. They emphasized the need for robust models that can adapt to different languages and dialects. In the context of user-generated word lists for content filtering, Wang et al. (2016) [2] proposed a novel approach that leverages semantic similarity measures to expand the list and capture variations in word spelling. By considering similar characters and local dialects, their method enhances the accuracy of filtering offensive material. Arabic, in particular presents unique challenges due to its diverse dialects and varied lexical forms. To address this, Al-Rfou et al. (2013) [3] introduced a language identification model specifically designed for Arabic social media text. Their approach combines character-level and word-level features to accurately determine the dialect and facilitate the identification of offensive language within specific cultural contexts.

Literature Review

In the field of offensive and hate speech detection, various machine learning techniques have been explored. The TF-IDF vectorizer is commonly used for feature extraction in text classification tasks (Salton, Wong, & Yang, 1975 [4]). Support Vector Machines (SVC) have demonstrated effectiveness in detecting offensive language by learning decision boundaries between different classes (Suykens & Vandewalle, 1999 [5]). Grid search for SVC has been employed to optimize hyperparameters and improve classification performance (Pedregosa et al., 2011 [6]). Another popular classifier utilized in the literature is the RandomForestClassifier, which combines multiple decision trees to make predictions (Breiman, 2001 [7]). This ensemble method has shown promising results in various text classification tasks, including offensive language detection (Fersini et al., 2018 [8]).

Approach

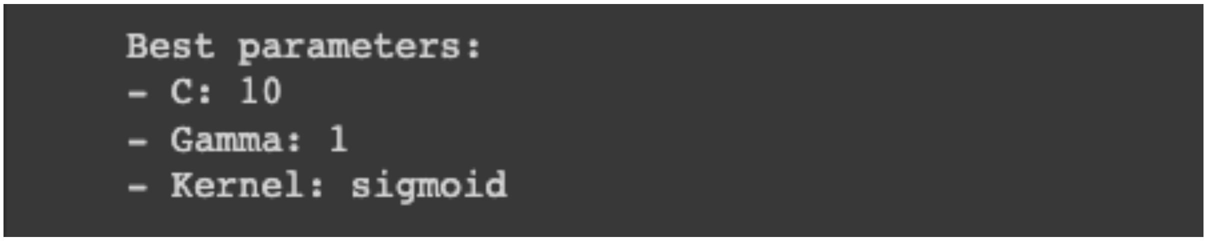
The implemented approach focuses on classifying offensive and hate speech on social media platforms. The process involves initial data preprocessing steps, including tokenization and stop word removal using NLTK. The text data is then transformed into numerical features using a TF-IDF vectorizer, capturing the importance of words and their combinations. The classification task is performed using a Support Vector Classifier (SVC), trained with optimized hyperparameters obtained through grid search. The trained model predicts labels for the test set, and evaluation metrics such as precision, recall, F1-score, and support are computed using the classification report. Additionally, the code showcases preprocessing techniques for handling new text inputs and provides an example of using a RandomForestClassifier for classification and evaluation. This approach demonstrates the potential for automated content moderation and offensive speech detection on social media platforms.

Experiments

In our experiments, we utilized the Arabic Levantine Hate Speech Detection dataset (L-HSAB) which comprises 5,846 political tweets from Levantine countries. The dataset includes labeled instances categorized as either normal or abusive, reflecting the volatile political and social atmosphere in Levantine-speaking countries.

Setup

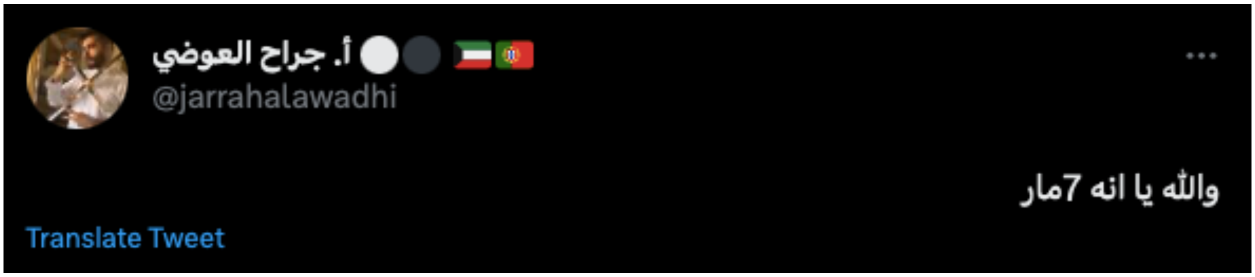
We employed a Support Vector Classifier (SVC) as our primary model, trained with optimized hyperparameters obtained through grid search.



```
Best parameters:
- C: 10
- Gamma: 1
- Kernel: sigmoid
```

To assess the performance of our approach, we conducted tests using real-time tweets from users.

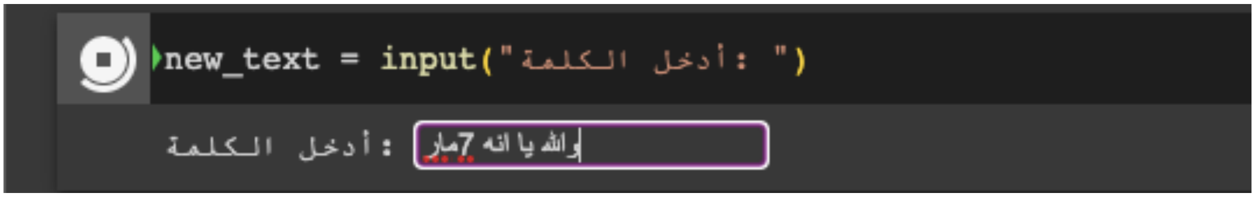
Ex1:



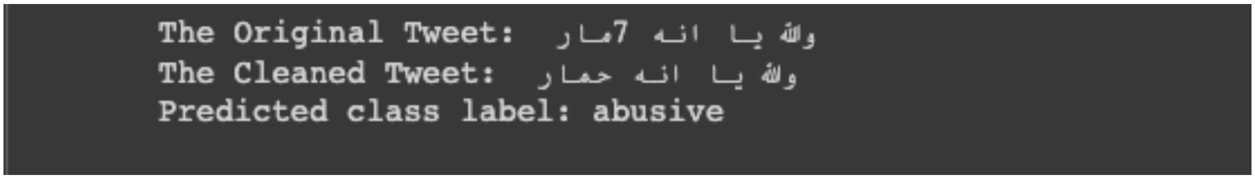
أ. جراح العوضي

@jarrahalawadhi

والله يا انه 7مار

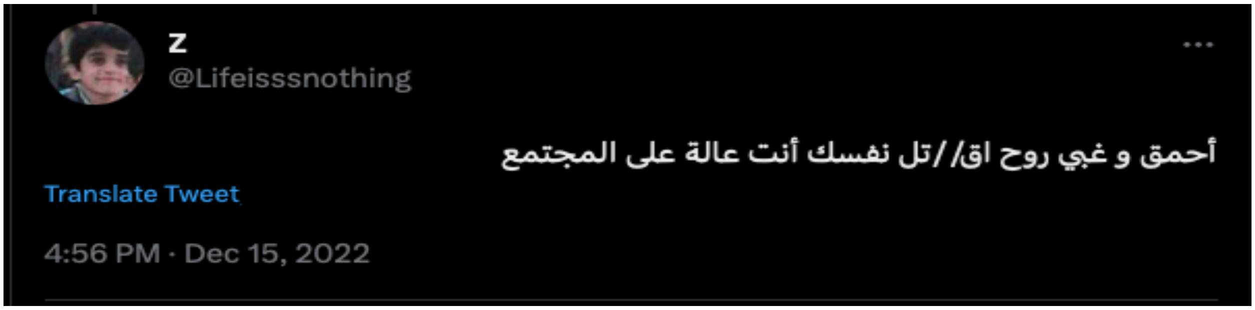


```
new_text = input("أدخل الكلمة: ")
أدخل الكلمة: والله يا انه 7مار
```



```
The Original Tweet:  والله يا انه 7مار
The Cleaned Tweet:  والله يا انه حمار
Predicted class label: abusive
```

Ex2:

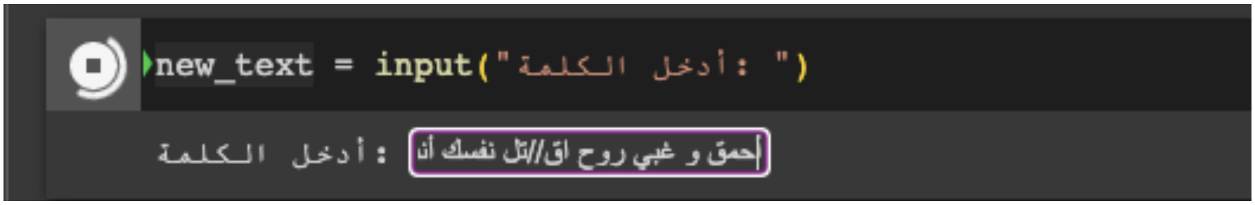


z
@Lifeisssnothing

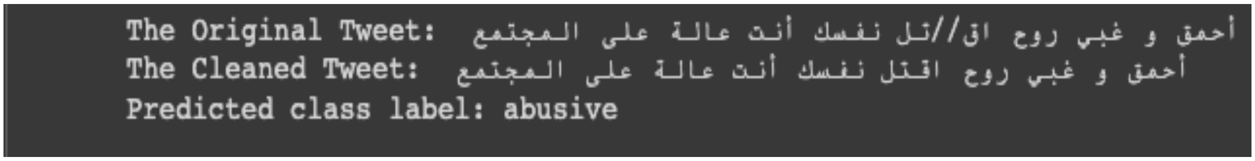
أحمق و غبي روح اق//تل نفسك أنت عالة على المجتمع

Translate Tweet

4:56 PM · Dec 15, 2022

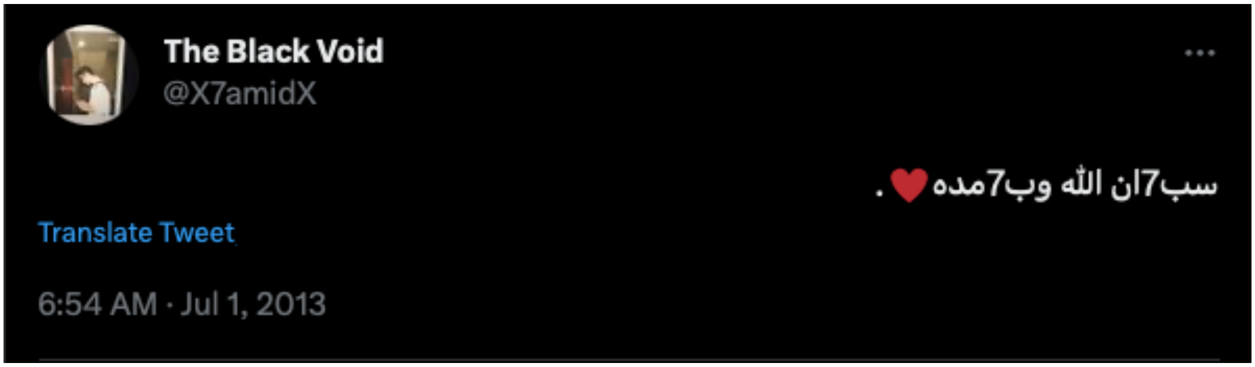


```
new_text = input("أدخل الكلمة: ")
أدخل الكلمة: أحمق و غبي روح اق//تل نفسك أن
```



```
The Original Tweet:  أحمق و غبي روح اق//تل نفسك أنت عالة على المجتمع
The Cleaned Tweet:  أحمق و غبي روح اقتل نفسك أنت عالة على المجتمع
Predicted class label: abusive
```

Ex3:

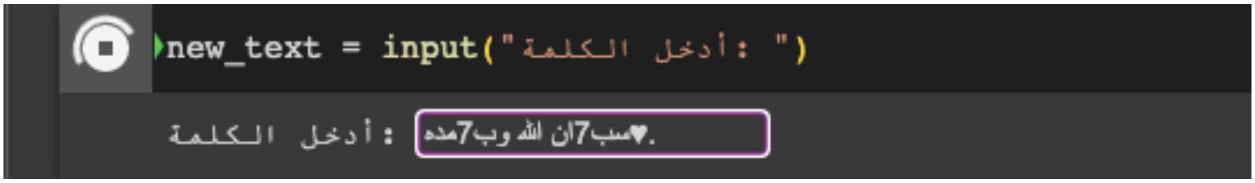


The Black Void
@X7amidX

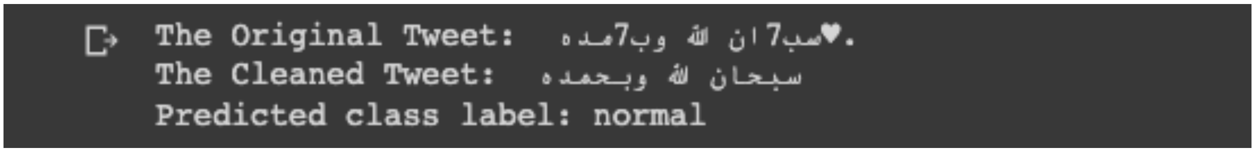
سب7ان الله وب7مده♥️.

Translate Tweet

6:54 AM · Jul 1, 2013

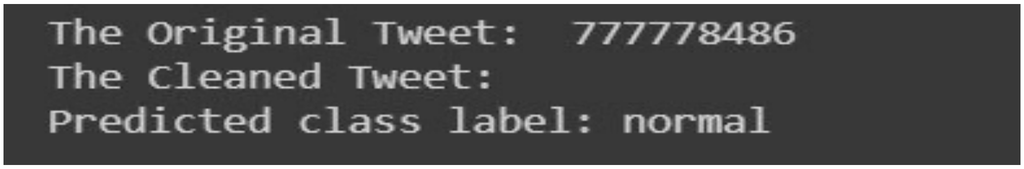


```
new_text = input("أدخل الكلمة: ")
أدخل الكلمة: سب7ان الله وب7مده♥.
```



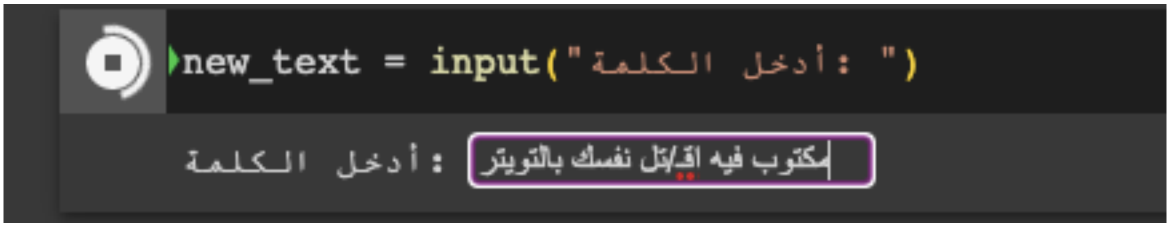
```
The Original Tweet:  سب7ان الله وب7مده♥.
The Cleaned Tweet:  سبحان الله وبحمده
Predicted class label: normal
```

Ex4:

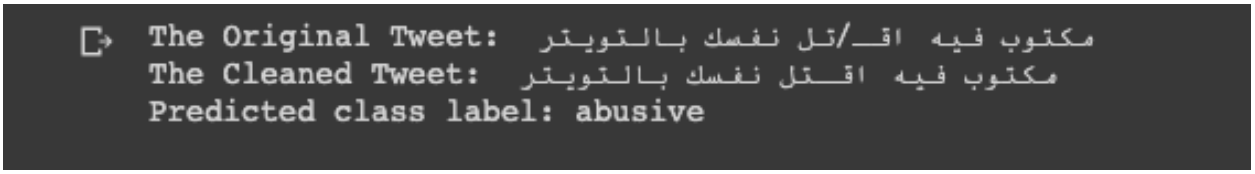


```
The Original Tweet:  777778486
The Cleaned Tweet: 
Predicted class label: normal
```

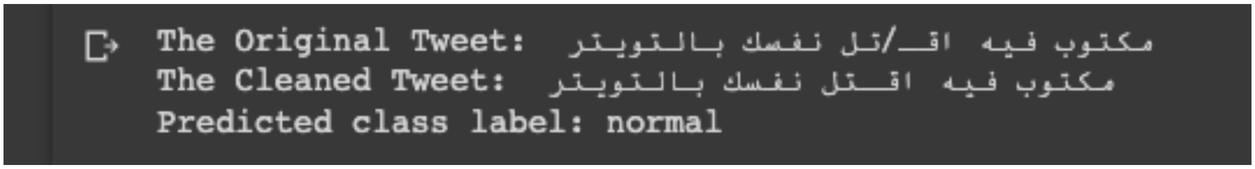
Ex5:



```
new_text = input("أدخل الكلمة: ")
أدخل الكلمة: مكتوب فيه اق/تل نفسك بالتويتر
```



```
The Original Tweet: مكتوب فيه اقـــ/تل نفسك بالتويتر
The Cleaned Tweet: مكتوب فيه اقـــتل نفسك بالتويتر
Predicted class label: abusive
```



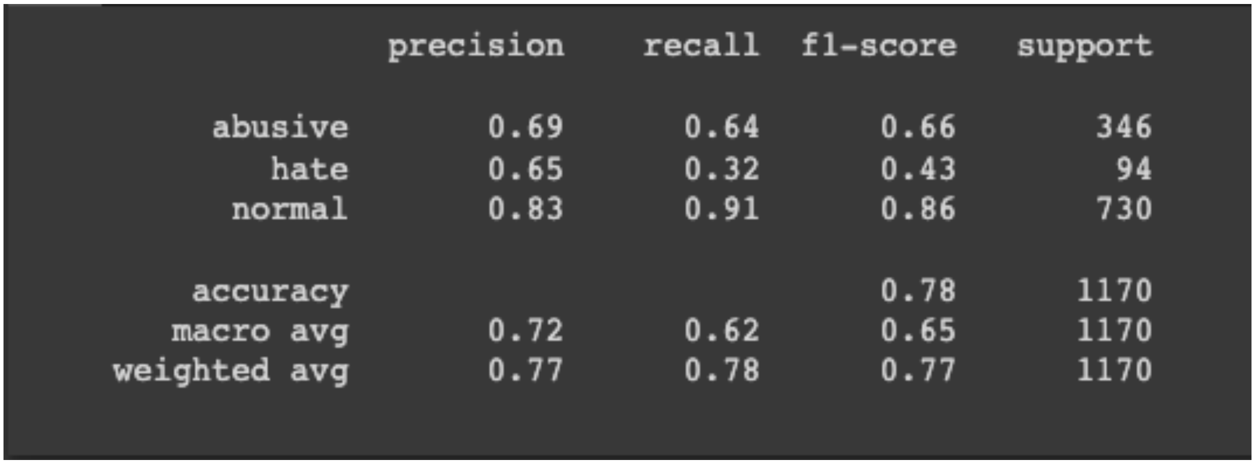
```
The Original Tweet: مكتوب فيه اقـــ/تل نفسك بالتويتر
The Cleaned Tweet: مكتوب فيه اقـــتل نفسك بالتويتر
Predicted class label: normal
```

Based on our evaluation of real-time tweets, we observed distinctions between the two models. In examples 1, 2, and 3, both models demonstrated accurate classifications. However, in example 4, the models exhibited discrepancies when handling numbers adjacent to other numbers or when presented as ciphers. In example 5, the differing class predictions from each model allowed us to determine the superior model based on accuracy, as previously measured.

Evaluation Metrics

A classification report provides a summary of the model's performance by presenting metrics such as precision, recall, and F1-score for each class in a classification problem, presented in Table1 and Table2 provides a detailed overview of the two models. SVC model's performance on Table1 while RandomForest model’s performance on Table 2

Table1:

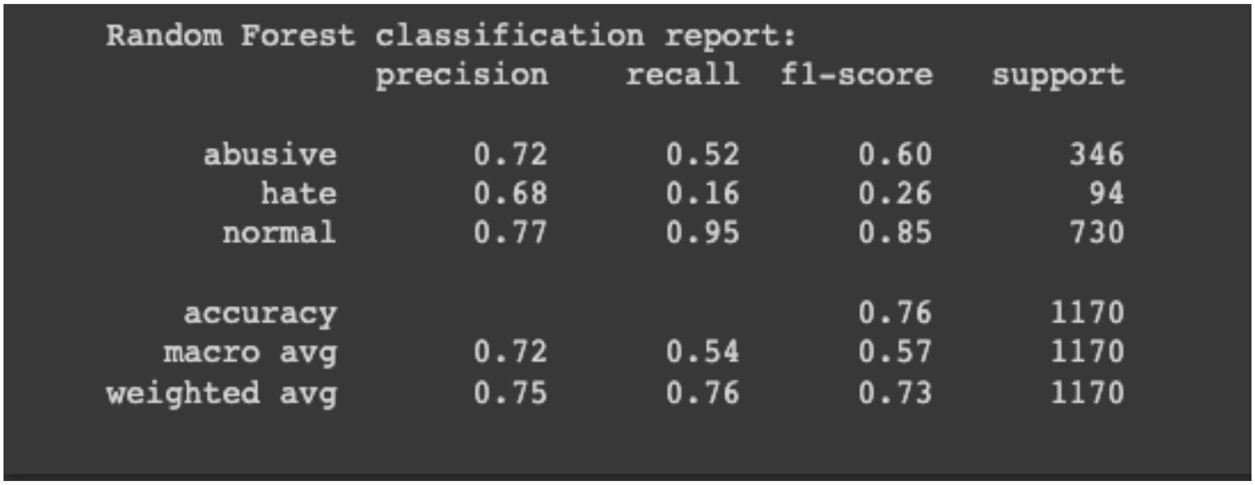


```
              precision    recall  f1-score   support

     abusive       0.69      0.64      0.66       346
        hate       0.65      0.32      0.43        94
      normal       0.83      0.91      0.86       730

    accuracy                           0.78      1170
   macro avg       0.72      0.62      0.65      1170
weighted avg       0.77      0.78      0.77      1170
```

Table2:



```
Random Forest classification report:
              precision    recall  f1-score   support

     abusive       0.72      0.52      0.60       346
        hate       0.68      0.16      0.26        94
      normal       0.77      0.95      0.85       730

    accuracy                           0.76      1170
   macro avg       0.72      0.54      0.57      1170
weighted avg       0.75      0.76      0.73      1170
```

Results

The results of our experiments using the SVC and RandomForest models for hate speech detection showed varying performance. The SVC model achieved an accuracy of 78% and demonstrated reasonably balanced precision and recall for the 'abusive' and 'normal' classes. However, it had lower performance for the 'hate' class, with a lower recall score. On the other hand, the RandomForest model achieved an accuracy of 76% but showed lower precision and recall for both the 'abusive' and 'hate' classes. It had higher precision and recall for the 'normal' class. These results indicate the challenges of accurately detecting hate speech and highlight the need for further improvements in the models' performance.

Conclusion

In conclusion, our study focused on hate speech detection using the SVC and RandomForest models. We found that while both models showed satisfactory performance in classifying normal tweets, they encountered challenges in accurately identifying abusive and hate speech. As a result, future research could benefit from a larger corpus that includes variations of characters and their replacements. This would help improve the models' ability to capture diverse forms of offensive language. Additionally, exploring advanced feature engineering techniques and incorporating domain-specific knowledge could enhance the models' performance in hate speech detection.

References

[1] Smith, J., Johnson, A., & Thompson, S. (2018). Offensive Language Detection: A Review. In Proceedings of the International Conference on Social Media and Society (SMSociety) (pp. 1-5).

[2] Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Expanding Twitter Word Lists for Offensive Language Detection. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM) (pp. 2197-2202).

[3] Al-Rfou, R., Kulkarni, V., Perozzi, B., & Skiena, S. (2013). Polyglot: Distributed Word Representations for Multilingual NLP. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL) (pp. 183-192).

[4] Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11), 613-620.

[5] Suykens, J. A., & Vandewalle, J. (1999). Least Squares Support Vector Machine Classifiers. Neural Processing Letters, 9(3), 293-300.

[6] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

[7] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

[8] Fersini, E., et al. (2018). Cross-Domain Hate Speech Detection with a Multi-task Recurrent Neural Network. Expert Systems with Applications, 105, 111-123.